THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

February 26, 1999

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.

APPLICATION NUMBER: *60/090,710*
FILING DATE: *June 26, 1998*

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

MARGARET BASSFORD
Certifying Officer

Express Mail Label NO. EL013268232US

A/PROV

JC604 U.S. PTO 06/26/98

JC541 U.S. PTO 60/090710 06/26/98

Please type a plus sign (+) inside this box →

PTO/SB/16 (12-97)
Approved for use through 1/31/98. OMB 0651-0037
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53 (c).

INVENTOR(S)		
Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
Paul Winfield Daniel Peter	Blacknell Denaro	Surrey, United Kingdom Hampshire, United Kingdom

☐ *Additional inventors are being named on the ___ separately numbered sheets attached hereto*

TITLE OF THE INVENTION (280 characters max)

SYSTEM AND METHOD FOR SHARING DOCUMENTS AMONG MULTIPLE PARTIES

CORRESPONDENCE ADDRESS

Direct all correspondence to:

☐ Customer Number → *Type Customer Number here*

OR

Place Customer Number Bar Code Label here

☒ Firm *or* Individual Name	DARBY & DARBY, P.C.				
Address	707 Wilshire Blvd., 32nd Floor				
Address					
City	Los Angeles	State	CA	ZIP	90017
Country	US	Telephone	(213) 243-8000	Fax	(213) 243-8050

ENCLOSED APPLICATION PARTS *(check all that apply)*

☒	Specification *Number of Pages* 10	☐	Small Entity Statement
☒	Drawing(s) *Number of Sheets* 5	☒	Other (specify) postcard receipt

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT *(check one)*

		FILING FEE AMOUNT ($)
☒	A check or money order is enclosed to cover the filing fees	
☒	The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number: 04-100	$150

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☒ No.

☐ Yes, the name of the U.S. Government agency and the Government contract number are: _____

Respectfully submitted,

SIGNATURE _____

TYPED or PRINTED NAME Stephen D. Burbach

TELEPHONE (213) 243-8000

Date 6/26/98

REGISTRATION NO. *(if appropriate)* 40,285

Docket Number: 0866-0E371US0

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Box Provisional Application, Assistant Commissioner for Patents, Washington, DC 20231.

SYSTEM AND METHOD FOR SHARING DOCUMENTS AMONG MULTIPLE PARTIES

BACKGROUND OF THE INVENTION

Many computer-related applications involve the use of shared data files. For example, in a multimedia (audio, data, video) conference application, a data file that includes a series of presentation slides may be transferred to computers used by the conference participants. Once transferred, the slides may be viewed by each of the participants. In some cases, the distributed data file may be edited by one or more of the participants.

In a typical on-line multi-party conference, a chairperson prepares a document (the conference document) that contains several data objects (data) such as Microsoft PowerPoint slides or Word documents. At the beginning of the conference, the chairperson connects to the other participants via a data network. Once connected, the participants may interact with one another in conjunction with the shared conference document. Inherent in this approach is that the conference document needs to be transmitted to the conference participants. Typically, the chairperson transfers the document to the other participants at the beginning of the conference. In practice, the file transfer process may take several minutes. As a result, the meeting participants may be forced to wait for the file transfers to complete before collaborating with one another or the chairperson must start the data exchange process early.

Conventionally, it is not sufficient to simply e-mail the conference document (which contains the data objects to be exchanged) to the participants in advance. This is because the objects themselves may change if they are edited before the meeting starts. In view of the above, a need exists for a file sharing method that reduces the amount of time that may be wasted transferring conference documents, but that also ensures that the participants in a conference are viewing the same versions of the data objects.

Additional problems arise in applications when multiple parties need to edit a shared document. In a Microsoft Windows environment, documents may be shared and edited using objects based on Microsoft's Object Linking and Editing ("OLE") technology. This technique also may be used for documents that are shared by computers that are connected via data networks.

For example, in a real-time conference involving several participants, the originator of the conference may embed the object into a suitable container document and then send the container document to the other participants. The object may then be edited by any of the participants. The edited object, in turn, may then be distributed to the other participants in the conference to update the object they are viewing to the current version.

This technique, however, often results in relatively long setup times when initiating the conference. In addition, relatively substantial data transfers may need to be made during the conference (i.e., large amounts of data may need to be distributed between the participants) whenever the objects are edited. In view of the above, a need exists for a method of document sharing that reduces the quantity of data sent during setup and that reduces data traffic during editing sessions of shared objects.

SUMMARY OF THE INVENTION

The invention provides a system and method for efficiently sharing data files among several parties. In accordance with one embodiment of the invention, a document to be used in a conference is routed to each of the participants before the conference is commenced. This may be accomplished, for example, using e-mail. In addition, each data object in the document is assigned a unique identifier and a digital signature that indicates when the object was last modified. When the chairperson connects to the conference, a list of identifiers and signatures is distributed to the participants. In the event a participant does not have the current versions of all the objects, that participant requests the corresponding new or altered objects. Significantly, the invention provides a method for only sending those objects that are not current (as opposed to the entire document) to the participant once the conference has begun. In many cases, the corresponding data may be sent in a fraction of the time that would have been needed to send the entire document. Accordingly, the invention may significantly reduce conference setup time.

Another embodiment of the invention provides a method of efficiently transferring conference documents that are defined in terms of presentation data and an entire data file. In this embodiment, only the presentation data is initially sent to the participants. The data file is then

transferred to a participant only when the participant wishes to edit the document. After the participant edits the document, updated presentation data is sent to the other participants.

In practice, the presentation data may comprise a quantity of data that is an order of magnitude less than the quantity of data in the data file (i.e., the entire document). As a result, a system constructed according to the invention may provide a significant reduction in the amount of data that is sent during the set up of the conference. Moreover, the data transmission process during an edit session may be more efficient in such a system. In summary, the invention may provide faster setup and editing time in comparison with conventional systems.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features of the invention will become apparent from the following detailed description of the preferred embodiments and the claims, when taken with the accompanying drawings, wherein similar reference characters refer to similar elements throughout and in which:

FIGURE 1 is a block diagram of one embodiment of a conferencing system constructed in accordance with the invention;

FIGURES 2A and 2B are a flowchart of one embodiment of a data sharing method in accordance with the invention;

FIGURE 3 is a flowchart of another embodiment of a data sharing method in accordance with the invention;

FIGURE 4 is a graphic illustration of an embedded method of data transfer; and

FIGURE 5 is a graphic illustration of one embodiment of a document sharing method in accordance with the invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

FIGURE 1 illustrates one embodiment of a conferencing system constructed according to the invention. The conferencing system includes several conference stations that are connected via one or more data networks (the Internet, a local area network, etc.). Conference participants (not shown) use the conference stations to communicate with other participants.

Each conference station may support audio, data and graphical services using various audio/video/data ("A/V/D") equipment. For example, the conference participants may speak with one another using the stations' audio equipment. A participant may route documents to other participants to enable the participants to simultaneously view the documents on the display devices (e.g., display monitors or electronic whiteboards) associated with the stations.

Each conference station runs a number of applications including multi-party conferencing applications. A conferencing application provides a user interface that facilitates establishing the conference and that enables a participant to route conference content information (e.g., documents, pictures, PowerPoint slides) to other participants. A given station also may include content editing/viewing applications that enable a participant to process the content information. For example, the content editing/viewing application may be a graphic-based editor such as Microsoft PowerPoint that enables a participant to edit graphical slides that are used in conjunction with the conference. To reduce the complexity of FIGURE 1, only a portion of the details of the stations are illustrated. In addition, the details shown for a given conference station might not be shown for all of the stations. It should be understood, however, that, in general, similar functionality is provided by each of the conference stations.

In accordance with one embodiment of the invention, the conferencing application includes a mechanism that enables a participant to efficiently distribute conference information (e.g., a conference document) to other participants. Specifically, the participant can send conference information to the other participants before the conference begins. Then, at the beginning of the conference, the participants receive electronic signatures (such as timestamps) indicative of the current version of the conference information. In the event a participant has an older version of the conference information, the participant can request an update from the originating station.

In accordance with another embodiment of the invention, the conferencing application includes a mechanism that enables a participant to efficiently distribute and edit conference information. Here, the conference information is associated with a set of presentation data and an entire data file. For example, a document that is compatible with Microsoft's Object Linking and Embedding (OLE) technology can be defined in terms of the data that defines the entire document

(the data file) and by another set of data (the presentation data) that merely defines, for example, how the document appears when it is displayed on a display screen.

In this embodiment, a participant can separately send the presentation data or the data file to the other participants. Thus, at the beginning of the conference, a participant or the conferencing application may elect to send only the presentation data to the other participants. In this case, only when a participant needs the entire file (for example, to edit the document) will the data file be sent to a participant. Typically, this technique will significantly reduce the amount of time it takes to initially send the conference information to the participants because, in general, there is much less presentation data than file data for a given document. Moreover, once the file has been edited, only the updated presentation data will need to be sent to many of the participants. Consequently, additional data transfer efficiencies may be realized by practicing this aspect of the invention.

Referring to FIGURES 2A, 2B and 3, the above operations will now be discussed in more detail. FIGURES 2A and 2B depict a flowchart of operations that may be performed in a conferencing system implemented in accordance with the first embodiment of the invention described above. In several of the examples below, it is assumed that the conference chairperson establishes the conference, and generates and distributes the documents that are used during the conference. It should be understood, however, that other parties (e.g., other conference participants) may perform these acts. It should also be understood that, in general, any of the participants may perform the caller and callee operations described in FIGURES 2A and 2B.

Initially, the conference chairperson generates the content information to be used during the conference and stores the information in a data memory such as a hard disk drive (FIGURE 2A, column 1). This content information may consist of various text or graphic files. For example, the content information may consist of a document generated using Microsoft Word, a slide presentation generated using Microsoft PowerPoint, a spreadsheet generated using Microsoft Excel, or a document generated by some other OLE-based application. Alternatively, the content information may consist of audio or video files (containing, for example, audio or video clips) that may be played on the participants' conference stations or other equipment.

The chairperson (or some other party) may create a conference document and insert various objects into the document (FIGURE 1). Using the PowerPoint document as an example, the chairperson may insert several PowerPoint slides into different pages of the conference document.

Assuming that each slide consists of 50 kbytes of data, it would take approximately six seconds to transmit each slide over a 64 kbit/second Wide Area Network link. Thus, it would take approximately four minutes to transfer a twenty slide presentation to two other participants.

In accordance with one embodiment of the invention, to prevent this delay from interfering with the conference, at some point in time prior to the conference, the participant who created the content sends the associated file to each of the other participants. In one embodiment, this is accomplished using e-mail. The file is then loaded by the recipient participants into their respective conference terminals.

In practice, the content information may be modified (e.g., by any of the participants) or new content information created after the conference document was sent to the conference participants (FIGURE 2A, column 1 or column 2). This scenario is accounted for in this embodiment through the use of identifiers and digital signatures as described below.

Each data object in the conference document may be associated with a unique identifier (such as a Globally Unique Identifier "GUID") and a signature that is used to ensure that the participants have the same version of the object (FIGURE 1). As FIGURE 1 illustrates, similar identifiers and/or signatures may be associated with the conference document as well. In general, an identifier is assigned to a document or an object when it is created. This identifier typically remains unchanged for at least the duration of the conference (even if the document or object is modified). In contrast, the signature assigned to a document or object is changed whenever the document or object is modified. For example, a document signature may be modified when any of its objects are modified (e.g., created, edited or deleted) and an object signature may be modified when the object is edited. In one embodiment, the signature consists of a timestamp that indicates the time at which an object was last modified.

Referring now to FIGURE 2B, the chairperson establishes the conference by invoking the mechanism by which the participants data network conferencing techniques are known in the art.

For example, a conferencing application such as INTERNET CONFERENCE PROFESSIONAL 2.1 sold by VocalTec Inc. of Northvale, New Jersey may provide conferencing over a data network using virtual conference rooms established on public or private network servers. The teachings of the invention may be integrated into a conferencing application such as this in order to improve the performance of the application.

As illustrated in FIGURE 1, some conferencing applications may use one or more conference bridges/servers to provide some of the conferencing functionality. For example, one or more of the participants may connect to a conference bridge. The conference bridge then performs the necessary operations to connect the streams (data, audio, etc.) of each of these (and perhaps other) participants.

Once the conference is established, the participant who generated the content information (e.g., conference document) sends a list of identifiers (e.g., the GUIDs of the documents and/or objects) and signatures (e.g., the timestamps of the documents and/or objects) to the other participants. Each participant (e.g., the conferencing application running on the participant's conference station) may then check to see whether the document/object versions of that participant's conference document match the document/object versions on the list. If the version do not match (e.g., an object has been changed or a new object was added after the document was sent), the participant with the old version may request the latest version of the object. In one embodiment, the corresponding request message includes the identifier and/or the signature of the desired document/object. Significantly, the participant may selectively request specific objects instead of the entire document, as necessary. Moreover, the document signature may be used to quickly verify that all of the objects are current or to request that the entire document be sent. Upon receipt of the request message, the participant or conferencing application in control of the document sends the requested document/objects to each requesting party in real-time.

From the above, it may be seen that the invention may significantly reduce the data transfer time at the beginning at the conference. In the case described above (20 slides distributed to two participants), the chairperson would send 40 information blocks that contain GUID and timestamp information. Assuming these blocks contain 100 bytes of data each, it is apparent that

the transmission time of this step will be negligible. Assuming the other participants loaded the prepared documents, the meeting can immediately proceed.

FIGURE 3 is a flowchart describing operations that may be performed in a conferencing system implemented in accordance with the second embodiment of the invention described above. In this example, it is assumed that the document to be shared has already been created and stored (for example, in a data memory).

For the purposes of describing this embodiment, the conference document is represented by two sections: presentation data and the data file (see FIGURE 1). In general, the presentation data consists of the minimum set of information that is required to display and/or print the document. In general, the data file consists of the minimum set of information that is required to edit/change the document.

An example of technology that supports the presentation data and data files as described herein is Microsoft's Object Linking and Editing ("OLE") technology. In the example that follows, it is assumed that the conference document is OLE compatible. Thus, the document definition includes presentation data that may be used by a client to display/print the document. In accordance with this embodiment of the invention, OLE is used in a linked manner so that a participant may initially distribute only the presentation data. The entire data file is sent to a participant only when that participant needs the entire file (e.g., when that participant wishes to edit the document).

A conference is established by connecting a group of conference participants (designated USERS A, B, C and D). This may be accomplished in a similar manner as discussed above.

A participant (e.g., USER A) then establishes a link to the document that is to be distributed to the other participants during the conference. That is, USER A places the document into the application container using linked presentation data only.

USER A then sends the presentation data to the other participants. As illustrated in FIGURE 1, a copy of the presentation data may be stored in the data memory in each participant's station. The participants may then view the document on their conference stations.

In one embodiment, the conferencing application may accomplish the above by intercepting the "paste" operation that USER A performs when she inserts objects into the

conference document. The conferencing application then automatically sends the presentation data to the other participants in the conference in real-time over the network. Each participant, then, may store a copy of the presentation data (e.g., as illustrated for conferencing station B in FIGURE 1).

At any time during the course of the conference (designated by the dashed line in FIGURE 3), a participant (e.g., USER B) may choose to edit the document. If the data file is not already available to that participant, it will be transferred on demand to him (and the data file stored in that his station, for example, as shown in station B in FIGURE 1). Next, the received file is linked to the original presentation data. The participant is then free to edit the document using a corresponding compatible content editing/viewing application (FIGURE 1) such as PowerPoint that is installed on his conference station.

As illustrated in FIGURE 3, different portions of the resulting modified file may be sent to different participants. The data file (with updates) is sent back to the originator of the file (e.g., USER A). Then, the originator updates the original file and its presentation object. As for the remaining participants (e.g., USERS C and D), they are only sent updates of the presentation data. The conferencing applications on the recipient terminals then automatically update the displayed document. This may be accomplished, for example, by sending GUIDs with the presentation data or data file, keeping track of which GUIDs are currently being displayed, and updating the display accordingly.

In practice, the presentation data may consist of much less data (e.g., ten times less data) than the data file (the total embedded file information). Thus, it may be seen that a system or method implemented according to this embodiment of the invention may be significantly more efficient than conventional document sharing methods. In addition to the efficiencies that may be obtained using all of the above techniques, significant efficiencies may be obtained by using only one of these techniques (for example, simply sending the deltas of the updates between the participants).

FIGURES 4 and 5 illustrate the advantages that may be realized through the use of this embodiment of the invention. FIGURE 4 illustrates data flow using an embedded method for sharing and editing documents. Here, USER A inserts an embedded object, USER B edits the

document, then the documents are updated for all of the participants. FIGURE 5 illustrates a linked method implemented in accordance with the invention. Here, USER A inserts an object via a link, the file is sent to USER B after USER B requests to edit the file, then all of the participants are updated. Data transfers (solid lines) indicate data moved before any editing occurs. The dotted lines represent data flows after a single remote point has edited the document. This example, assumes that the presentation data is 1/10 the size of the embedded data and that subsequent deltas are 1/10 of the original size of either the embedded data or the presentation data.

TABLE 1 summarizes the data flow in each case. It may be seen that the method of the invention can significantly reduce the total data transmitted over the networks, particularly when less than all of the members of the conference will edit a particular document. In particular, data sent during the conference setup may be dramatically reduced. In addition, data transmission during an edit session may be more efficient. Thus, the invention may provide faster setup and editing time, particularly on relatively slow data networks.

Data compression techniques may also be used to further reduce the amount of data that is transferred for the conference. For example, the data may be compressed when the changes for the presentation of the full object are being sent.

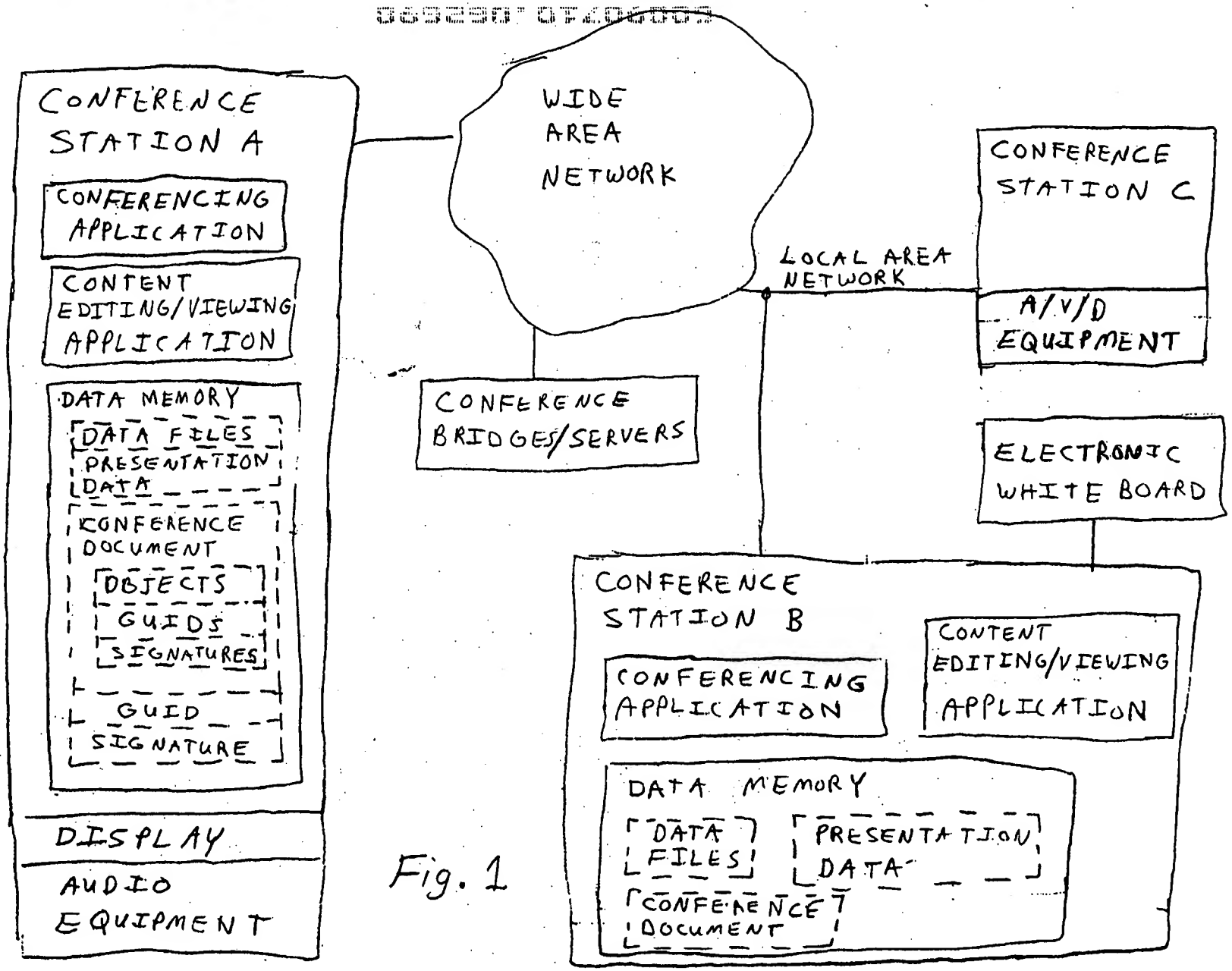
CONFERENCE STATION A
CONFERENCING APPLICATION
CONTENT EDITING/VIEWING APPLICATION
DATA MEMORY
DATA FILES
PRESENTATION DATA
CONFERENCE DOCUMENT
OBJECTS
GUIDS
SIGNATURES
GUID
SIGNATURE
DISPLAY
AUDIO EQUIPMENT
WIDE AREA NETWORK
CONFERENCE BRIDGES/SERVERS
LOCAL AREA NETWORK
CONFERENCE STATION C
A/V/D EQUIPMENT
ELECTRONIC WHITE BOARD
CONFERENCE STATION B
CONFERENCING APPLICATION
CONTENT EDITING/VIEWING APPLICATION
DATA MEMORY
DATA FILES
PRESENTATION DATA
CONFERENCE DOCUMENT

Fig. 1

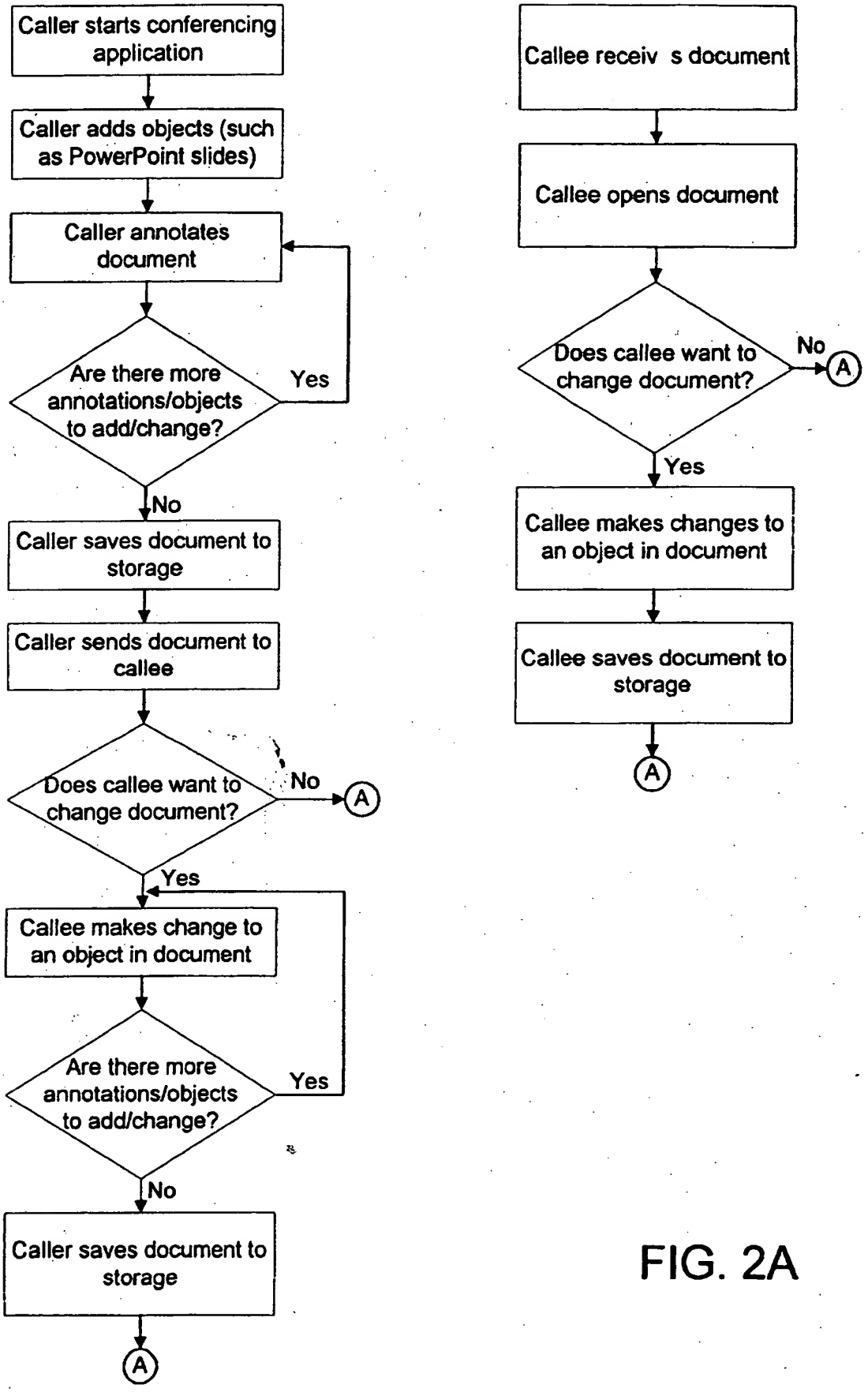
Caller starts conferencing application
Caller adds objects (such as PowerPoint slides)
Caller annotates document
Are there more annotations/objects to add/change?
Yes
No
Caller saves document to storage
Caller sends document to callee
Does callee want to change document?
No
A
Yes
Callee makes change to an object in document
Are there more annotations/objects to add/change?
Yes
No
Caller saves document to storage
A
Callee receiv s document
Callee opens document
Does callee want to change document?
No
A
Yes
Callee makes changes to an object in document
Callee saves document to storage
A

FIG. 2A

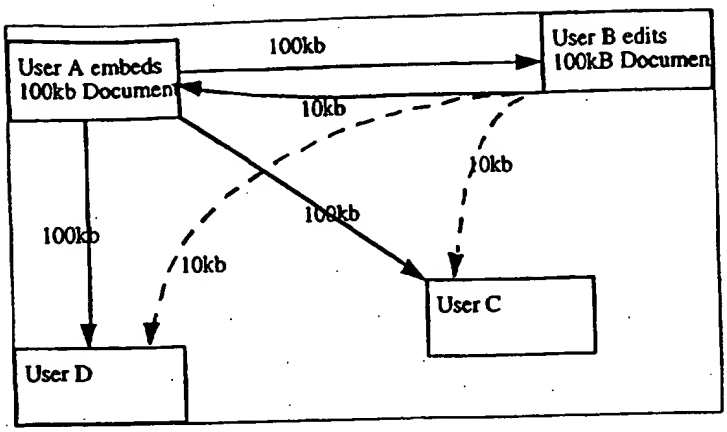

Fig. 4

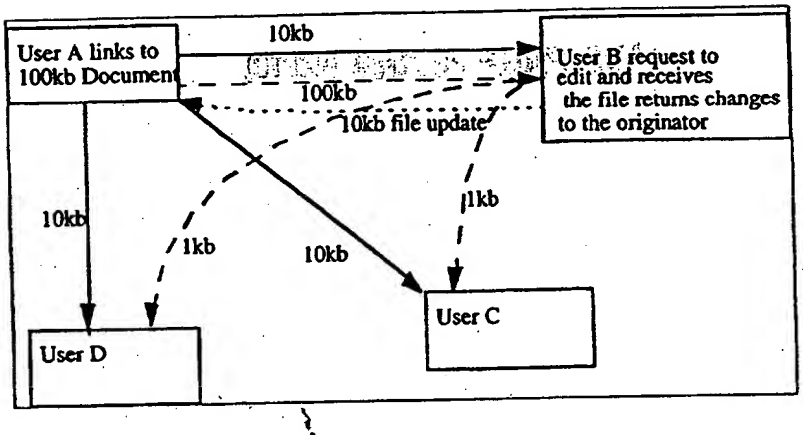

Fig. 5

	Embeded data (Fig. 4) Total data sent in kilobytes	*Fast linked method* (Fig. 5) Total data sent in kilobytes
Setup data	300	30
B Edits	330	142
B+C Edits	360	254
B+C+D Edits	390	366

TABLE 1

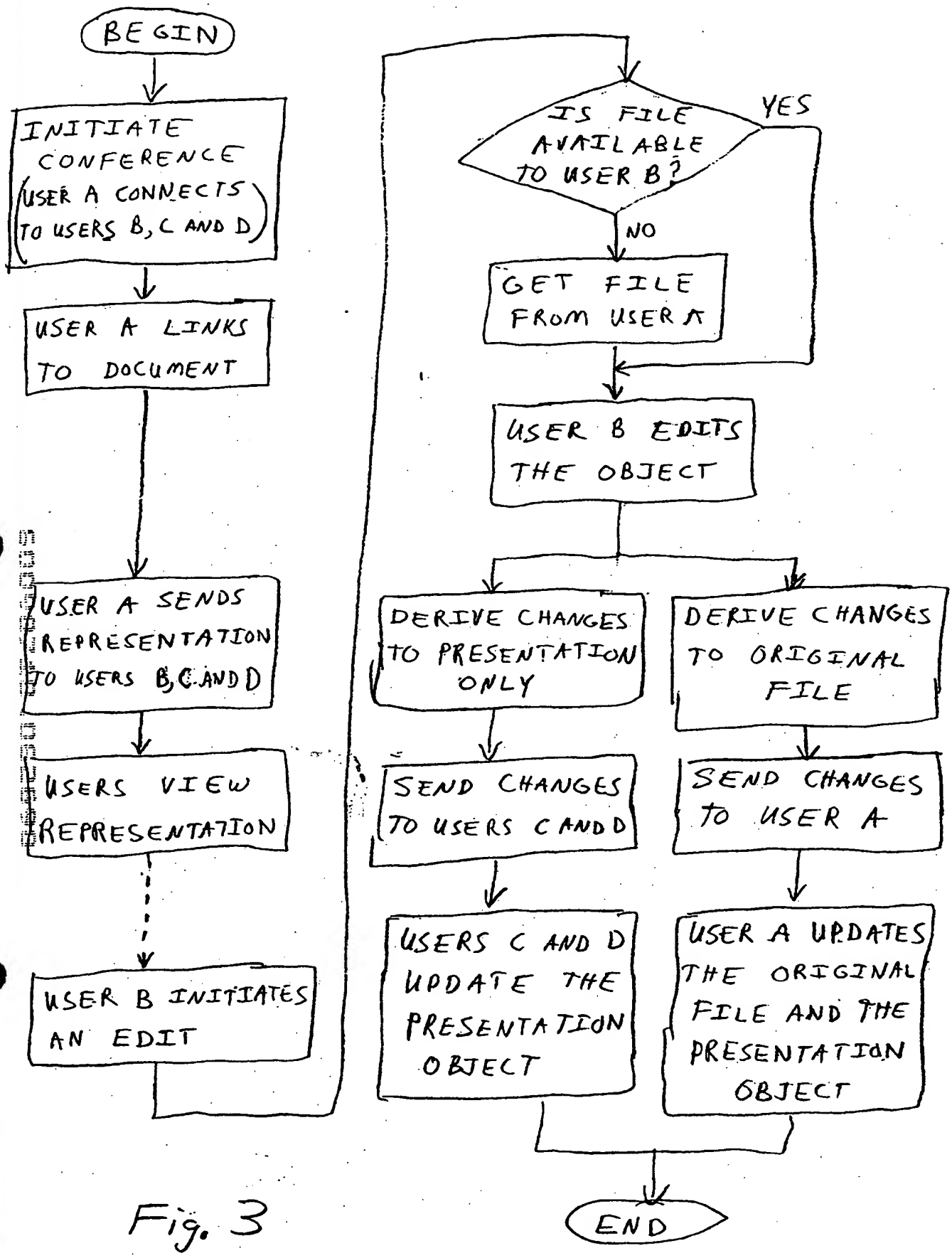
BEGIN
INITIATE CONFERENCE (USER A CONNECTS TO USERS B, C AND D)
USER A LINKS TO DOCUMENT
USER A SENDS REPRESENTATION TO USERS B, C AND D
USERS VIEW REPRESENTATION
USER B INITIATES AN EDIT
IS FILE AVAILABLE TO USER B?
YES
NO
GET FILE FROM USER A
USER B EDITS THE OBJECT
DERIVE CHANGES TO PRESENTATION ONLY
DERIVE CHANGES TO ORIGINAL FILE
SEND CHANGES TO USERS C AND D
SEND CHANGES TO USER A
USERS C AND D UPDATE THE PRESENTATION OBJECT
USER A UPDATES THE ORIGINAL FILE AND THE PRESENTATION OBJECT
END

Fig. 3

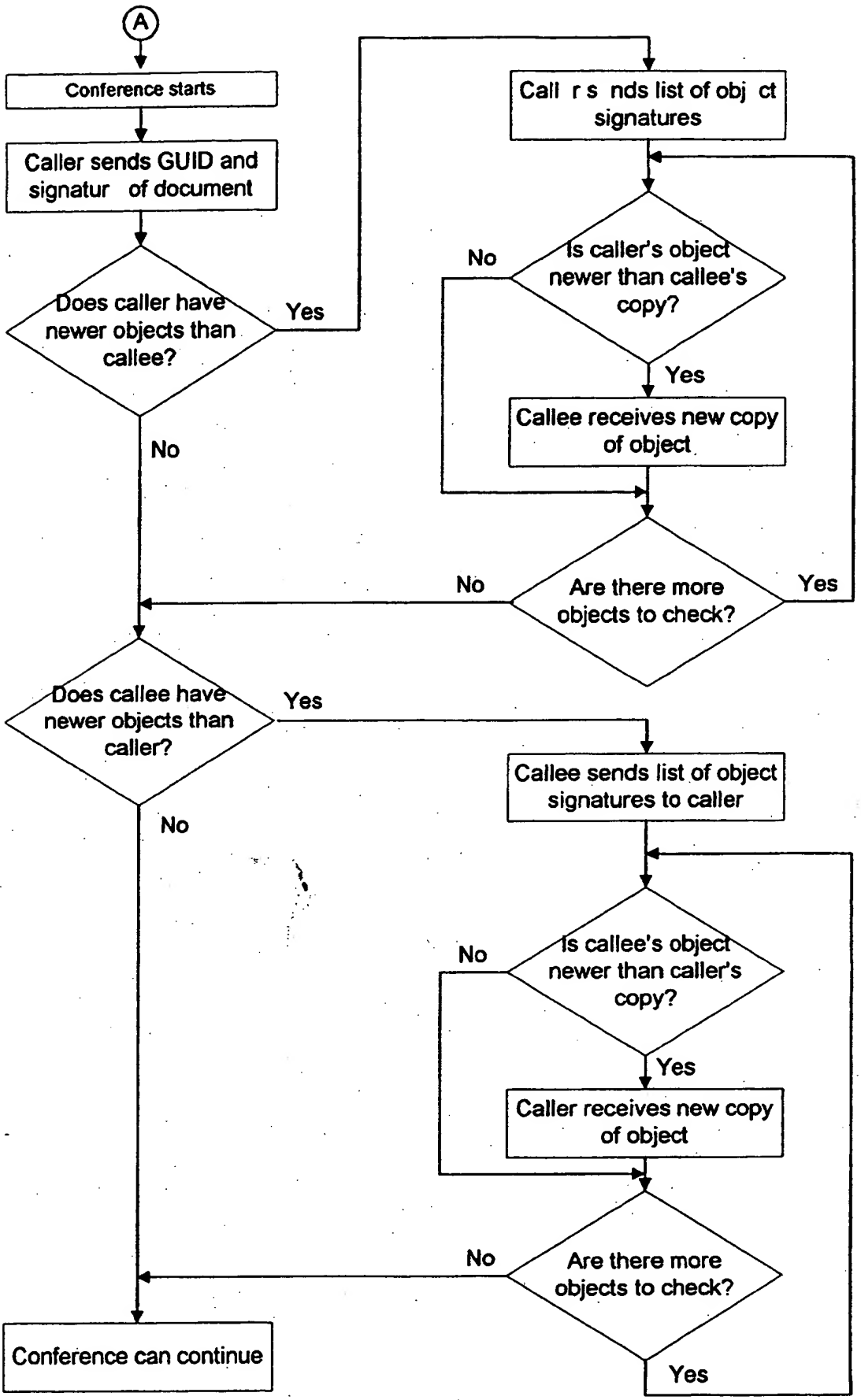
A
Conference starts
Caller sends GUID and signatur of document
Does caller have newer objects than callee?
Yes
No
Call r s nds list of obj ct signatures
Is caller's object newer than callee's copy?
No
Yes
Callee receives new copy of object
Are there more objects to check?
No
Yes
Does callee have newer objects than caller?
Yes
No
Callee sends list of object signatures to caller
Is callee's object newer than caller's copy?
No
Yes
Caller receives new copy of object
Are there more objects to check?
No
Yes
Conference can continue

FIG. 2B